

# AIA – Integration von Systemen

**Tjark Bikker**  
**PROMATIS software GmbH**  
**Hamburg**

## **Schlüsselworte:**

Application Integration Architecture (AIA), AIA Foundation Pack, Lifecycle Demo

## **Einleitung**

Oracle Application Integration Architecture (AIA) stellt ein auf der Oracle SOA Suite aufbauendes Framework dar, das die Implementierung von Integrationsprojekten über den Entwicklungszyklus gesehen stark vereinfacht.

AIA stellt verschiedene Werkzeuge zur Verfügung, die in jeder Phase der Entwicklung den Integrationsarchitekten, den Entwickler oder den Systemadministrator bei ihren Aufgaben unterstützen. Gleichzeitig wird durch die Verwendung dieser Werkzeuge auch eine gewisse Konformität der Application Integration Architecture erzwungen, was die reibungslose Verbindung der verschiedenen Komponenten ermöglicht.

Zusätzlich zur Architektur und den Werkzeugen bietet das AIA Foundation Pack eine umfangreiche Sammlung von vordefinierten Geschäftsobjekten, die aufgrund eines durchdachten Konzepts einfach für spezielle Bedürfnisse angepasst werden können.

Dieser Vortrag vermittelt einen groben Überblick über den Entwicklungszyklus mit AIA und stellt anhand eines einfachen Beispiels die Verwendung der verschiedenen Werkzeuge bei der Entwicklung vor.

## **Die Application Integration Architecture**

Im Folgenden werden die Kernkomponenten der Application Integration Architecture im Überblick beschrieben.

- ABM: Application Business Message
- EBO: Enterprise Business Object
- EBM: Enterprise Business Message
- ABCS: Application Business Connector Service
- EBS: Enterprise Business Service

## **Transportadapter**

Transportadapter transportieren gegebenenfalls die Daten aus der Applikation in das Application Business Message-Format, falls die Applikation keine oder eingeschränkte Möglichkeiten bietet, um direkt mit der SOA-Ebene per Webservice zu kommunizieren.

## **ABM – EBO – EBM**

Das frei definierte XML-Format, in dem die jeweilige Nachricht aus der Application in den SOA-Layer transportiert wird, wird Application Business Message (ABM) genannt.

Für die Weiterleitung der ABM in die Application Integration Architecture wird die ABM in das sogenannte Enterprise Business Object (EBO) umgewandelt. Dieses EBO enthält die zu übertragenden Daten für die jeweilige Business Task.

Ein EBO wird in eine Enterprise Business Message (EBM) verpackt, die zusätzlich zu den reinen Daten Informationen beispielsweise darüber enthält, welche Aktion auf den Daten ausgeführt werden soll, von welchem System die Nachricht schon verarbeitet worden ist und an welche Systeme die Nachricht weitergeleitet werden soll.

### ABCS

Applikationen werden über sogenannte Application Business Connector Services (ABCS) in die SOA-Landschaft eingebunden. Falls in dem Business Case ein Service in Anspruch genommen werden soll, spricht man von einem Requestor. Falls ein Service angeboten werden soll, wird der Service Provider genannt.

Der Requestor nimmt Nachrichten im ABM-Format an und hat die Aufgabe, die Informationen aus der ABM anzureichern und zu validieren, sie in eine EBM zu übertragen und an einen Enterprise Business Service (EBS) zu übertragen.

### EBS

Der EBS übernimmt für ein bestimmtes EBO das Weiterleiten der Nachrichten. Je nach Komplexität des Prozesses sendet der EBS die Nachricht direkt weiter an einen ABCS-Provider oder an in der Architektur höher angesiedelte Konstrukte, beispielsweise den Enterprise Business Flow (EBF). Diese Business Flows orchestrieren, falls notwendig, die Verwendung verschiedener EBS.

In Abbildung 1 ist der Kommunikationsfluss zwischen den AIA-Komponenten abgebildet. Die äußersten Elemente sind die sogenannten Transportadapter, die die Application Business Messages an die jeweiligen Application Business Connector Services übergeben.

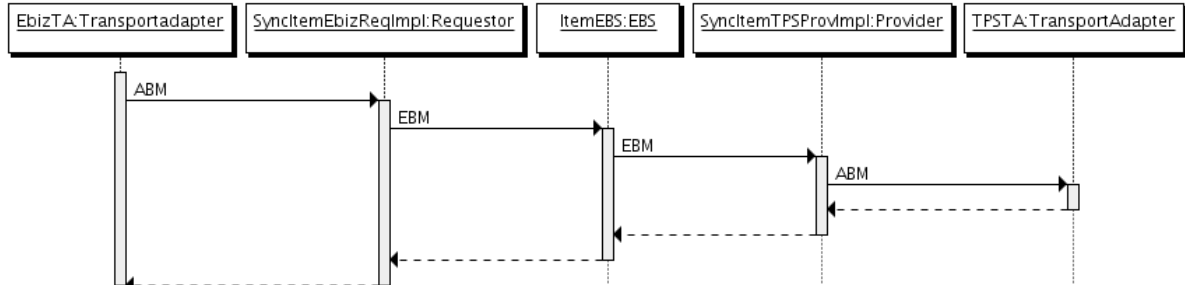


Abbildung 1: Kommunikation der AIA-Komponenten

### Der AIA Lifecycle

Im Folgenden soll der Durchlauf einer AIA-gestützten Entwicklung beschrieben werden. Als Beispiel für die Integration von zwei Systemen soll hier die einfache Übertragung von geänderten Lieferantendaten von der E-Business Suite an ein Legacy-System demonstriert werden.

Bevor die AIA-gestützte Entwicklung beginnen kann, müssen selbstverständlich die Schnittstellen spezifiziert werden. Sobald die Spezifikation abgeschlossen ist, kann die erste Phase des AIA Lifecycles beginnen.

### Dekomposition

Die erste Phase des AIA Lifecycles ist die funktionale Dekomposition. In diesem Schritt analysiert der Systemarchitekt die Spezifikationen und identifiziert die Komponenten, die für die einzelnen Projekte gemäß der Application Integration Architecture implementiert werden müssen.

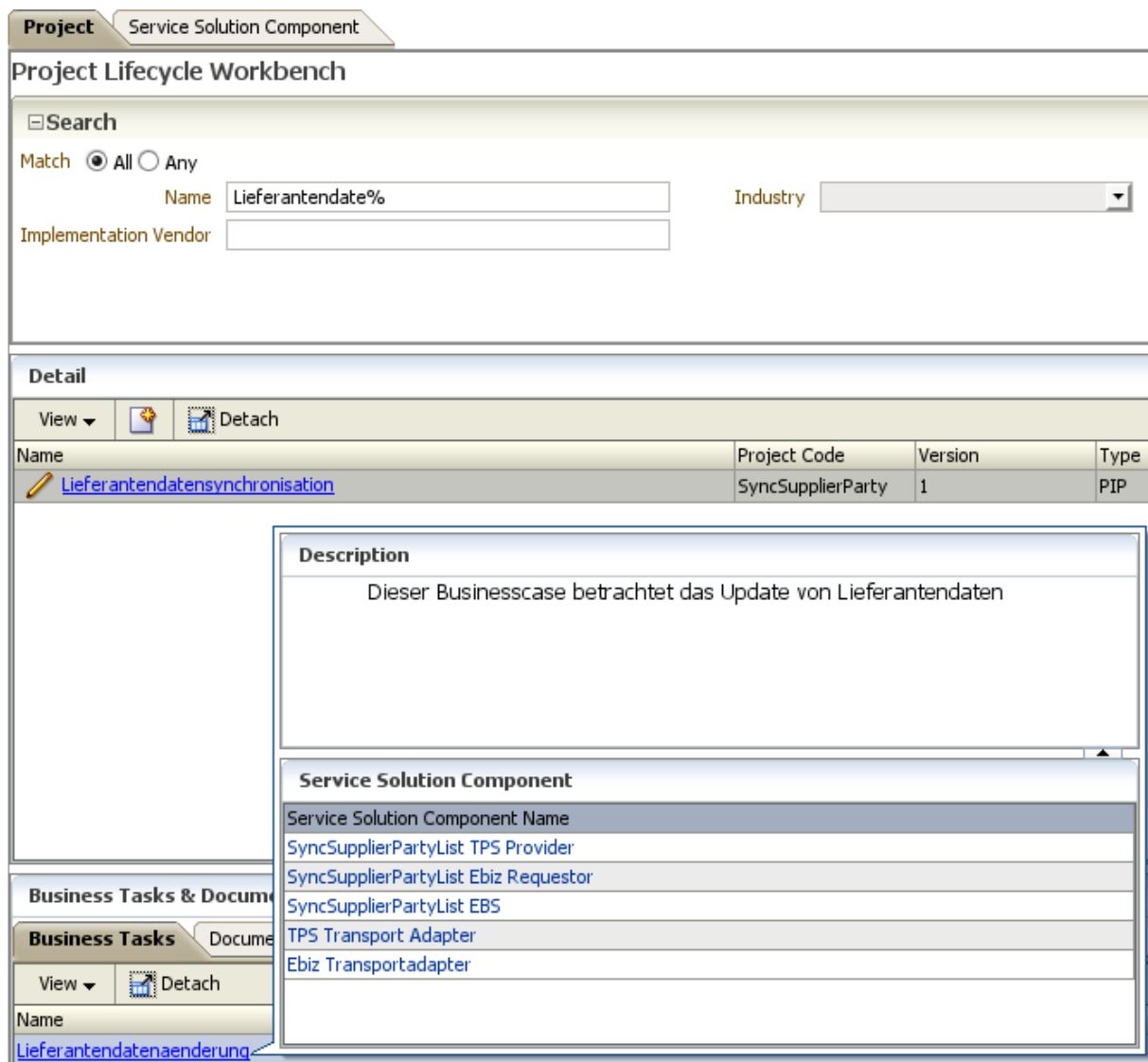


Abbildung 2: AIA Project Lifecycle Workbench

Das vom AIA-Framework für diesen Zweck bereitgestellte Werkzeug ist die AIA Lifecycle Workbench. Dieses Werkzeug bietet dem Systemarchitekten die Möglichkeit, verschiedene Projekte einzurichten, die die jeweiligen Implementierungen repräsentieren.

Ein einzelnes Projekt kann hierbei verschiedene Business Tasks beinhalten. Für das betrachtete Beispiel gibt es das Projekt "Lieferantendatensynchronisation", mit der einzigen Business Task "Lieferantendatenaenderung". Ein weiterer denkbarer Business Case wäre "Lieferantendatenneuanlage".

Innerhalb des Business Cases werden nun vom Systemarchitekten die einzelnen Service Solution Components definiert.

Im Beispiel sind folgende Service Solution Components zu definieren:

- Transportadapter Ebiz
- Lieferantendatenaenderung Requestor Ebiz
- Enterprise Business Service Lieferantendaten

- Lieferantendatenaenderung Provider TPS
- Transportadapter TPS

### **Entwicklung**

Nachdem die Service Solution Components definiert worden sind, können sie von den Entwicklern implementiert werden.

Für die Entwicklung der Requestor- und Provider-Komponenten steht der Service Component Wizard als Extension für den JDeveloper bereit. Dieser Wizard erstellt einen passenden BPEL-Prozess für die jeweilige Komponente.

In diesem automatisch erstellten BPEL-Prozess ist das Error handling in einer Form aufgebaut, dass jeder nicht behandelte Fehler über das AIA Error handling verarbeitet wird. Auch die Inhalte von Pflichtfeldern in der Enterprise Business Message werden im Requestor den AIA-Vorgaben entsprechend vorbelegt. Da der Wizard keine Informationen darüber haben kann, wie die Transformation von der Application Business Message auf die Enterprise Business Message aussehen soll, muss die Transformation selbstverständlich angepasst werden. Durch den XSL-Editor des JDevelopers ist dies im Regelfall eine recht einfache Aufgabe.

Für eine korrekte Funktion der erstellten Composites müssen im Anschluss noch die Interface-Definition, die WSDL-Location und der Binding Port überprüft werden.

Falls die Komponenten getestet werden sollen, bevor die Installation mit dem AIA Installation Driver vorgenommen wurde, muss der Entwickler einige zusätzliche Installationsschritte manuell vornehmen:

- Informationen aus der automatisch generierten AIAServiceConfigurationProperties.xml in die AIAConfigurationProperties.xml im Metadata Storage eintragen.
- Dateien in den Metadata Storage laden.
- AIA Konfiguration neu laden

Als letzten Schritt in der Entwicklung sollten die erstellten Komponenten annotiert werden, damit Sie im nächsten Schritt verarbeitet werden können. Bei den automatisch erstellten Komponenten (Requestor/Provider) sind diese Annotations schon automatisch generiert worden.

### **Harvesting**

Nachdem die Entwicklung abgeschlossen ist können die Implementierungsdetails geerntet werden.

Dabei werden die Informationen, die in der ersten Phase in der Lifecycle Workbench eingetragen wurden, durch Details aus der Implementierung angereichert.

Es gibt grundsätzlich zwei verschiedene Ziele für das Harvesting, das Oracle Enterprise Repository (OER) und die AIA Lifecycle Workbench. Da das OER eine optionale Komponente ist, wird hier nur das Harvesting in die Lifecycle Workbench betrachtet. Das Harvesting-Script unterstützt jedoch ebenso wie das Harvesting in die Lifecycle-Workbench, das Harvesting nur in das OER oder in beide Komponenten gleichzeitig.

Der einfachste Weg um das Harvesting durchzuführen ist, die entwickelten Projekte auf dem AIA-Server im \$AIA\_HOME abzulegen. Damit das Harvesting-Script die erstellten Dateien finden kann, muss noch die Datei mit den Harvester Settings konfiguriert werden.

Nachdem die Konfiguration durchgeführt wurde, kann der Harvesting-Prozess mit dem Befehl

```
$AIA_HOME/Infrastructure/LifeCycle/AIAHarvester/AIAHarvest.sh
-partial true -mode AIA -settings HarvesterSettings.xml
```

gestartet werden. Für die Verwendung unter Windows ist ein ähnliches Script unter dem Namen AIAHarvest.bat vorbereitet.

## Deployment

Der letzte Schritt im AIA Lifecycle ist das automatische Deployment mit dem AIA Installation Driver (AID). Um das Deployment durchführen zu können, muss zunächst aus den Informationen aus der AIA Lifecycle Workbench eine Bill of Material (BOM) generiert werden.

Leider gibt es in der Lifecycle-Workbench die Schwelle, dass die Projekt-Dekomposition nicht mehr bearbeitet werden kann, nachdem die BOM einmal generiert worden ist. Daher sollte dieser Schritt gut überlegt sein. Zwar kann die generierte BOM wieder gelöscht werden, allerdings ist das ein von Oracle nicht unterstützter Eingriff.

Ist die Bill of Material einmal generiert, werden in ihr die als Service Solution Component für das fragliche Projekt identifizierten Komponenten aufgelistet. Falls aufgrund fehlender Globally Unique Identifier (GUID) in den Annotationen die geernteten Komponenten nicht in der BOM aufgeführt werden, können weitere Komponenten manuell hinzugefügt werden.

Aus dieser Bill of Material kann dann mit dem AIA Deployment Plan Generator ein Deployment Plan erstellt werden, der die zur Installation auf einem neuen Server notwendigen Informationen enthält.

```
#>cd $AIA_HOME/utils/DeploymentPlanGenerator
#>ant --noconfig -Dinput=BOM.xml -DdeploymentPlan=DP.xml -
DharvesterSettings=DP_HS.xml
```

Nachdem der Deployment Plan generiert worden ist, kann mit dem AIA Installation Driver der Deployment-Prozess auf dem Zielsystem durchgeführt werden.

```
Ant -f DeploymentDriver.sh -Dasd
#>ant -DdeploymentPlan=DP.xml
-DpropertiesFile=InstallationProperties.xml
-f
$AIA_HOME/Infrastructure/Install/AID/AIAInstallDriver.xml
```

Während des automatischen Deployments werden alle Artefakte (Data Sources, Connection Factories, JMS-Queues etc.) angelegt, die für den Betrieb der implementierten Komponenten notwendig sind.

Auch die vorher erwähnten manuellen Setupsschritte (MDS-Upload, Service-Konfiguration) werden automatisch durchgeführt.

Da das Erstellen einiger Artefakte im Zweifelsfall einen Neustart erfordert, wird dieser Neustart auch automatisch von dem Installation Driver über die WLST-Konsole vorgenommen.

## Kontaktadresse:

### Tjark Bikker

PROMATIS software GmbH  
Notkestraße 9  
D-22607 Hamburg

Telefon: +49 (0) 40 253 32 69 - 0  
Fax: +49 (0) 40 253 32 69 - 99  
E-Mail: [tjark.bikker@promatis.de](mailto:tjark.bikker@promatis.de)  
Internet: [www.promatis.de](http://www.promatis.de)